

Herramientas para colaborar en la red:

SCM

Nicolás Ruiz

<nicolas@ula.ve>

Investigación y Desarrollo

Centro de Tecnologías de Información

Corporación Parque Tecnológico de Mérida

Organización de la Charla

- Introducción a los SCM
- Descripción de CVS
- Nuevos requerimientos de SCMs modernos
- Alternativas
- Una opción: Monotone
- Demostración

¿Qué significa SCM?

- Software Configuration Management
- Software Control Management
- Source Control Management
- Source Configuration Management

Es una herramienta que permiten controlar cambios en la configuración de un sistema.

En el contexto que nos compete nos referimos a código fuente, documentos, estructuras de directorios.

¿Cuál es el problema?

Cuando varias personas desarrollan código (o documentos) - sobre todo durante un periodo de tiempo no trivial - se vuelve progresivamente más difícil poder tener una idea clara de lo que está ocurriendo.

Los SCM nos permiten llevar un control sobre el proceso de desarrollo de una manera que facilita el responder preguntas y completar acciones habituales durante dicho proceso de desarrollo.

Para que sirve un SCM

- Mantener respaldo de los archivos (todas las versiones).
- Permitir acceso a los archivos (todas las versiones).
- Avisar cuando múltiples usuarios hacen cambios incompatibles a los mismos archivos.
- Responder a las siguientes preguntas:
 - ¿Cuál es la última versión del archivo X?
 - ¿Quién fue el último en modificar el archivo X?
 - ¿Cuáles fueron los cambios hechos al archivo X el 30 y 31 de febrero?

Para que NO sirve un SCM

- Acelerar el proceso de desarrollo en un proyecto que está críticamente falto de tiempo.
- Generar una historia que no existe.
- Decidir como reconciliar múltiples versiones incompatibles a un documento (esto es tarea de los usuarios del SCM).
- Imponer restricciones en las modificaciones.
- Decidir que debe mantenerse en el repositorio.

Usar un SCM implica más trabajo que no usarlo. Su valor consiste en simplificar las cosas en un futuro.

Como se usa un SCM

0. Crear un repositorio para un proyecto y se colocan archivos en el. Posiblemente hecho por terceras personas.
1. Hacer una copia local de la data.
2. Manipular (modificar, agregar, eliminar, mover) la data.
3. Subir la data modificada al repositorio.
4. Copiar del repositorio a la copia local data modificada por otros.
5. Goto 2

Ejemplos de lo valioso de los SCM

- Después de desarrollar código por unos meses, necesito las versiones anteriores (benchmarking, bugs).
- Se necesita reestructurar un documento en el que se sigue trabajando (branches).
- Se agrega una nueva persona a un grupo que está desarrollando activamente un proyecto (history).

CVS: el SCM de facto

- CVS: Concurrent Versions System.
- Es el sistema de control de versiones más popular actualmente.
- Es el sistema usualmente utilizado para permitir acceso público a software activamente desarrollado.
- Es una serie de scripts que utilizan RCS a bajo nivel.
- Código abierto, cuenta con soporte comercial.
- Binarios disponibles para Linux, Windows, Java.
- Múltiples front-ends.

Ventajas de CVS

- Estable.
- Bien documentado.
- Bien conocido (es facil encontrar ayuda).
- Funciona bien.
- Interfaz sencilla (?).
- Soporta múltiples sistemas operativos.
- Fuente abierta (GPL v1).

Limitaciones de CVS

- Las operaciones con el repositorio no son atómicas.
- No soporta archivos binarios.
- Pobre manejo de renombrado/copiado de archivos y directorios (se pierde la historia).
- Sistema centralizado.
- Más vale comenzar a etiquetar desde el principio, y bien.
- Pobre manejo de permisologías?

Nuevos requerimientos de los SCM

- Transacciones con los repositorios deben ser atómicas.
- Manejo de archivos binarios.
- Comprensión de renombrado/copiado de archivos.
- Seguridad (acceso restringido a porciones del repositorio).
- SCM distribuidos.
- Cherry picking.
- Changesets.

Alternativas (Propietarios)

- Aegis
- Arch
- Bazaar
- Bazaar-NG
- BitKeeper
- ClearCase
- CMSynergy
- Co-Op
- Darcs
- git
- Mercurial
- Monotone
- OpenCM
- Perforce
- PureCM
- Subversion
- Superverision
- svk
- Vesta
- Virtual SourceSafe

Monotone

- Distribuido.
- Sistema de almacenamiento sencillo (SQL).
- Basado en transacciones.
- Soporta Linux, Windows, OS X, otros UNIXes.
- Scriptable (Lua).
- Eficiente protocolo de sincronización (netsync).
- Versiones basadas en firmas criptográficas (SHA1).
- Clientes usan llaves RSA para autenticarse.
- GPL.
- Sin dependencias externas.

Más ventajas de monotone

- Entiende renombrar/mover archivos y directorios.
- Mantiene la historia cuando se copian/mueven archivos.
- Se pueden replicar los repositorios sin problema.
- Bien documentado.

Limitaciones de Monotone

- Internacionalización no es completa (nombres de archivos, certificados, claves tienen que ser ASCII).
- En pleno desarrollo.
- No puede eliminar directorios del repositorio (hay que borrar los archivos individualmente).
- No tiene cliente gráfico.
- No tiene cliente Web.
- El soporte para “cherry picking” no es el mejor.

Enlaces

- Better SCM Initiative. <http://better-scm.berlios.de/>
- Version Control Systems Comparison:
<http://better-scm.berlios.de/comparison/comparison.html>
- SCM on Linux: <http://linuxmafia.com/faq/Apps/scm.html>
- Comments on OSS/FS SCM:
<http://www.dwheeler.com/essays/scm.html>
- RevCtrl wiki: <http://revctrl.org/FrontPage>
- monotone: <http://www.venge.net/monotone/>
- Bazaar-NG: <http://www.bazaar-ng.org/>
- Subversion: <http://subversion.tigris.org/>